Practical - 3

Array and arithmetic operations

[A]

1.) Create PHP Script that calculates Simple Interest for the specified: Amount, Period, and Rate of Interest. Display Amount, Period, and Rate of Interest and calculated Simple Interest.

2.) Create PHP Script

   a. To convert “47” to its binary value using decbin()

   b. To convert “11001101101.101010” to its decimal value using bindec ().

[B]

Write PHP program to create two arrays of students. The first one have roll no and name, the second one have roll no and marks. Make a single array having three values for each student, roll no, name and marks. Add two elements using [] syntax. Using iterator functions, process each element and display with row number and all the attribute values for all records. Extract them in variables myID, myName and myMarks using suitable functions.

[C] Execute the following code snippets and write the output for it, identify errors if any :-

1.

```
<?php
        $browsers = array ("Firefox", "Internet Explorer", "Opera");
        echo "<select>";
        foreach($browsers as $browser)
        {
                echo "<option name='$browser'>$browser</option>";
        }
        echo "</select>";
?>
```

2.

```
<?php
$articles = array
(
 "PHP Variables" => "A variable is a mean to store values
 such as strings or integers so we can easily
 reuse those values in our code...",

 "PHP Strings" => "A string is a sequence of letters, symbols,
 characters and arithmetic values...",
 "PHP Loops" => "In programming, we often repeat an action or a
 piece of code a number of times using loops
 to solve a problem..."
);
echo "<table border='1'>";
foreach ($articles as $article_title => $article_body)
{
        echo "<tr>";
        echo "<td>";
        echo $article_title;
        echo "</td>";

        echo "<td>";
        echo $article_body;
        echo "</td>";
        echo "</tr>";
}
echo "</table>";
?>
```

3.

```
<?php
        echo "<table width='100' align='center'>";
        for($i=0; $i<=5; $i=$i+1)
        {
                if($i % 2 == 0)
                {
                        echo "<tr>";
                        echo "<td style='background-color:red'>";
                        echo $i;
                        echo "</td>";
                        echo "</tr>";
                }
                else
                {
                        echo "<tr>";
                        echo "<td style='background-color:green'>";
                        echo $i;
                        echo "</td>";
                        echo "</tr>";
                }
        }
        echo "</table>";
?>
```

4.

```
<?php
$seven = 7;
$arrayname = array( "this is an element", 5, $seven );
```

	echo $arrayname[0]; //prints: this is an element
	echo $arrayname[1]; //prints: 5
	echo $arrayname[2]; //prints: 7
	?>

5.	<?php $first_array = array("key1" => "the first element", "key2" => "the second element"); $second_array = array("key3" => "this is the first element of the second array", "key4" => "this is the second element of the second array",); echo $first_array['key1']; echo $second_array['key3']; echo $first_array['key2']; echo $second_array['key4']; ?>

6.	<?php
	$flower_shop = array (
	     "rose" => "5.00",
	     "daisy" => "4.00",
	     "orchid" => "2.00"
	);
	echo "rose costs $flower_shop['rose'], daisy costs $flower_shop['daisy'], and orchild costs $flower_shop['orchild'].";
	?>

7.	<?php
	$flower_shop = array (
	     "rose" => "5.00",
	     "daisy" => "4.00",
	     "orchid" => "2.00",
	);
	echo "<table border='1' cellpadding='5'>";
	echo"<tr><th>Flower</th><th>Price</th></tr>";
	foreach($flower_shop as $Flower=>$Price)
	{
	  echo "<tr><td>$Flower </td><td>$Price</td></tr> ";
	}
	echo "</table>";

	?>
8.	<?php $flower_shop = array("rose" => array("5.00", "7 items", "red"), "daisy" => array("4.00", "3 items", "blue"), "orchid" => array("2.00", "1 item", "white"),); echo "rose costs ".$flower_shop['rose'][0].", and you get ".$flower_shop['rose'][1]."."; echo "daisy costs ".$flower_shop['daisy'][0].", and you get ".$flower_shop['daisy'][1]."."; echo "orchid costs ".$flower_shop['orchid'][0].", and you get ".$flower_shop['orchild'][1]."."; ?>
9.	Sorting Numerically Indexed Array :- <?php $flowers = array("rose", "daisy" , "orchid", "tulip", "camomile"); sort($flowers); for ($i=0; $i <= 4; $i++) echo $flowers[$i]."<br\>"; ?>

10.	<?php
	$shop = array ("rose" => "5.00",
	                "daisy" => "4.00",
	                "orchid" => "2.00",
	              );
	asort($shop);
	foreach($shop as $key => $value)
	  echo $key." costs ".$value." dollars<br />";
	?>

11.	<?php
	$shop = array ("rose" => "5.00",
	                "daisy" => "4.00",
	                "orchid" => "2.00",
	              );
	ksort($shop);
	foreach($shop as $key => $value)
	  echo $key." costs ".$value." dollars<br />";

```
?>
```

12.

```
<?php
$try[] = array("11", "12", "15", "22", "41", "42");
$try[] = array("6", "7", "16", "17", "22", "23");
$count = count ($try);
for ($i=0; $i<$count; $i++){
            $countmore=count($try[$i]);
            for ($j=0; $j < $countmore; $j++){
                        print ("i$i j$j " . $try[$i][$j] . "<br> ");
            }
            print ("<br>");
}
?>
```

13.

```
<?php
$array1=$array3=array(1,2,3,4,5,6,
array('new delhi','mumbai','kolkata'));
echo"<br/><b>Value of original array is:</b><br/>";
print_r($array1);
$array2=array('a','b','c');
array_push($array1,$array2);
echo"<br/><b>After appending an array:</b><br/>";
print_r($array1);
$var="new variable";
echo"<br/><b>After appending a value:</b><br/>";
array_push($array1,$var);
print_r($array1);
echo"<br/><b>After appending to itself :</b><br/>";
array_push($array3,$array3);
print_r($array3);
?>
```

14.

```
<?php
$array1 = array("10", 100, 100, "a");
$array2 = array(1, 3, "2", 1);
array_multisort($array1, $array2);
print_r($array1);
print_r($array2);
?>
```

15.

```
<?php
  $marks = array(
                        "Saurav" => array
                        (
                        "physics" => 35,
                        "maths" => 30,
                        "chemistry" => 39
                        ),
                              " Rahul " => array
                (
                "physics" => 30,
                "maths" => 32,
                "chemistry" => 29
                ),
                "Veer" => array
                (
                "physics" => 31,
                "maths" => 22,
                "chemistry" => 39
                )
                    );
   echo "Marks for Saurav in physics : " ;
   echo $marks[' Saurav ']['physics'] . "<br />";
   echo "Marks for Rahul in maths : ";
   echo $marks['Rahul ']['maths'] . "<br />";
   echo "Marks for Veer in chemistry : " ;
   echo $marks['Veer']['chemistry'] . "<br />";
?>
```

16. range() :- The range() function creates an array containing a range of elements.

```
<?php
$number = range(0,5);
print_r ($number);

$number = range(0,50,10);
print_r ($number);

$letter = range("a","d");
print_r ($letter);

?>
```

17. array_keys(array,value) :-

Parameters	Description
Array	Required. Specifies an array
Value	Optional. You can specify a value, then only the keys with this value are returned
strict	Optional. Used with the value parameter. Possible values:

	• true - Returns the keys with the specified value, depending on type: the number 5 is not the same as the string "5". • false - Default value. Not depending on type, the number 5 is the same as the string "5".

```
<?php
$a=array("a"=>"Horse","b"=>"Cat","c"=>"Dog");
print_r(array_keys($a));

$a=array("a"=>"Horse","b"=>"Cat","c"=>"Dog");
print_r(array_keys($a,"Dog"));

$a=array(10,20,30,"10");
print_r(array_keys($a,"10",false));

$a=array(10,20,30,"10");
print_r(array_keys($a,"10",true));
?>
```

18.

```
<?php
$a=array("Dog","Cat");
print_r(array_pad($a,5,0));

$a=array("Dog","Cat");
print_r(array_pad($a,-5,0));
?>
```

19. The rsort() function sorts an array by the values in reverse order. This function assigns new keys for the elements in the array. Existing keys will be removed. This function returns TRUE on success, or FALSE on failure.

```
<?php
$my_array = array("a" => "Dog", "b" => "Cat", "c" => "Horse");

rsort($my_array);
print_r($my_array);
?>
```

20. The krsort() function sorts an array by the keys in reverse order. The values keep their original keys.This function returns TRUE on success, or FALSE on failure.

```
<?php
$my_array = array("a" => "Dog", "b" => "Cat", "c" => "Horse");
krsort($my_array);
print_r($my_array);
?>
```

21. The ksort() function sorts an array by the keys. The values keep their original keys. This function returns TRUE on success, or FALSE on failure.

```
<?php
$my_array = array("a" => "Dog", "b" => "Cat", "c" => "Horse");
ksort($my_array);
print_r($my_array);
?>
```

22.

```
<?php
   $author = "j@java2s.com";
   $author = str_replace("@","(at)",$author);
   echo "Contact the author of this article at $author.";
?>
```

23.

```
i) <?php
   echo str_replace("world","Peter","Hello world!");
  ?>

ii) <?php
    $arr = array("blue","red","green","yellow");
    print_r(str_replace("red","pink",$arr,$i));
    echo "Replacements: $i";
    ?>

iii) <?php
      $find = array("Hello","world");
     $replace = array("B");
     $arr = array("Hello","world","!");
     print_r(str_replace($find,$replace,$arr));
     ?>
```

24. strpos() :- The strpos() function returns the position of the first occurrence of a string inside another string.If the string is not found, this function returns FALSE.

Syntax :- strpos(String s ,String find, int start)

Parameter	Description
String s	Required. Specifies the string to search.
String find	Required. Specifies the string to find.
int start	Optional. Specifies where to begin the search

```
<?php
$newstring = 'abcdef abcdef';
$pos = strpos($newstring, 'a', 1);

?>
```